Journal of Accessibility and Design for All

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087 DOI: 10.17411/jacces.v4i3.49

# AN ACCESSIBLE INTERFACE FOR PROGRAMMING AN ASSISTIVE ROBOT

Juan G. Victores[1], Santiago Morante[2], Alberto Jardón Huete[3], Carlos Balaguer[4]

*ORCID: 0000-0002-3080-3467[1], 0000-0003-4933-7694[2], 0000-0002-3734-7492[3], 0000-0003-4864-4625[4]*

Robotics Lab research group, Universidad Carlos III de Madrid (Leganés) Spain

jcgvicto@ing.uc3m.es[1], smorante@ing.uc3m.es[2], ajardon@ing.uc3m.es[3], balaguer@ing.uc3m.es[4]

***Abstract*:** In this paper, we present an accessible interface in the context of our work on bringing advanced robotics closer to everyday domestic users. This interface allows inexperienced users to be capable of programming an assistive robotic arm to perform a specific desired task in a household environment. The programming process is performed through the developed Web Browsable interface, within which a Task Creator Wizard plays an essential role. The robot's open architecture enables flexible multi-modal interaction. In addition to the touch buttons provided by the Web Browsable interface when presented on a touch screen, voice commands and the use of the Wii Remote™ controller for intuitive robotic movement have also been enabled. The Web Browsable interface has been designed to provide high accessibility while taking aesthetic details into account, in order to prevent distraction caused by boredom of the user.

***Keywords:*** Assistive Robot, Graphical Interface, Usability, User-Centered Design.

## Introduction

In our everyday lives, we are increasingly being surrounded by modern technologies. Advanced electronics are exponentially embedded in

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

smartphones, tablet PCs, ebooks. Inexperienced people and even young children are able to interact with touch screens or buttons, navigating through tabs, menus, and icons (Holzinger, 2003). This fact provides the reason for existence of end-user developments (EUD) (Burnett & Scaffidi, 2011). With all the possibilities for efficiency and improvement of EUD, some researchers have already begun to see the potential of web applications (Rode, Rosson, & Qui, 2006) and alternative controllers (Guo & Sharlin, 2008). Web interfaces can additionally provide several potential benefits, such as ubiquitous availability, and public access if desired.

All of these advances are also being progressively incorporated in the field of robotics, although perhaps at a slower pace. Robotics and automation are fields that are first commercially developed for industrial environments, with non-friendly interfaces, requiring technical training for personnel. However, recent works such as Baxter (Guizzo & Ackerman, 2012) are now taking the user-oriented point of view into account, aiming at simplifying industrial manipulator programming.

From the steady regime of production plants, robotics and automation technology can now be found in retail stores, and ultimately, in home environments. In their broadest scope, robotics and automation include everything from motorized shutters and vacuums to less common advanced robotic manipulators (Iocchi, Ruiz-del-Solar, & Van der Zant, 2012). Current worldwide research focuses on how to introduce dynamic and mobile elements to perform "household chores" and daily tasks that require complex manipulation and advanced reasoning skills. These technologies will begin to make our life easier only with the development of human-robot interfaces that provide comfort and satisfaction to the user (Kim, Oh, Choi, Jung, & Kim, 2011). In this paper, the authors propose the merger of robotics with technology that everyday users can be familiar with, such as web browsing, voice commanded control, and video-game controllers, and present proof-of-concept Open Source implementations and documentation with experimental results.

These developments have taken place using an assistive robotic arm, which is currently located in an assistive living kitchen test environment. The following is a review of some of the assistive robotic arm's most important features and characteristics.

- Full on-board robot control and communications, with no need for an external control cabinet.
- Unlimited workspace through power supply climbing connectors.
- Light-weight symmetrical structure for climbing.
- Tool exchange system for grippers, utensils, sponge, etc.
- Portable and friendly interfaces adapted to different levels of user diversities and preferences.
- Open architecture for flexible component integration.

These last two features are the ones that have been mostly exploited by the authors in this paper, in order to extend the reach of their developments to the hands of everyday home domestic users.

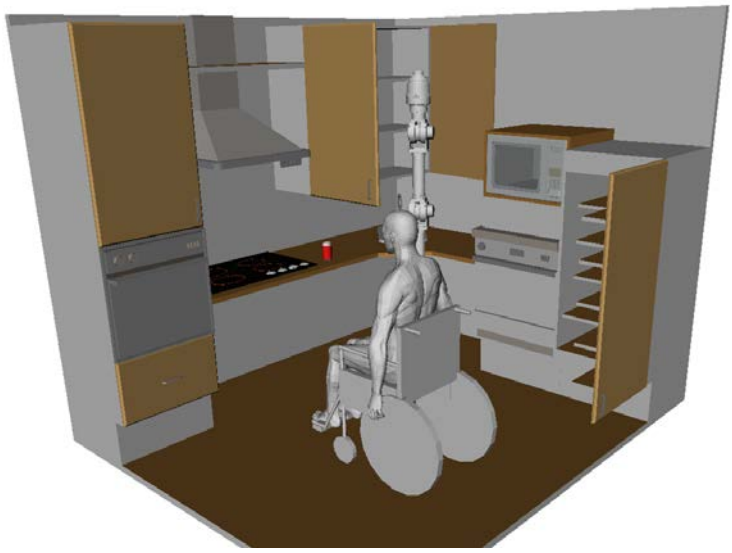
## Methodology

To provide an accessible interface for a system as complex as an assistive robotic arm, an open architecture must be provided. We provide this open architecture through the use of the YARP robotics platform (Fitzpatrick, 2008), that enables multi-modal interaction by providing a flexible and robust implementation of the publisher/subscriber paradigm. This platform is lightweight enough for our embedded system, and provides multi-lingual and multi-platform support combined with easiness of use for a great range of possible developer profiles (Victores, 2010). The developed YARP modules are intended to a run within the Wireless Local Area Network (WLAN) of the robot, but the user is free to expose the interface connections for external assistants to collaborate, remotely interacting with the modules from a distant location.

A simulator environment is crucial for an initial training phase, to allow end users to practice with the assistive robotic arm before handling it in the real

physical environment. The simulator used is OpenRAVE (Diankov, 2010), given that it is lightweight, modular, and exposes an application programmer interface to its core libraries. The default environment it is set to load is the robot's assistive living kitchen test environment (Fig. 1). Robot sensors and cameras are also incorporated in the simulated 3D environment to provide a higher degree of realism and fidelity.

*Figure 1. The simulator is set to load the robot's assistive kitchen model*



One of the main objectives of our assistive robotic arm research and software development of the past years has been to provide integrated modes of Human-Robot Interaction (HRI) through devices with which users can already be previously acquainted with, therefore allowing them to immediately start discovering how to control the robot platform through the interface devices instead of using time learning how to use a new specific interface device. The robot system's open architecture methodology additionally allows the control interface devices to be used simultaneously. These device modules are all managed coherently by the system.

## Touch Buttons

The robot's Web Browsable interface is intended for display on devices that support tactile interaction. It is composed by nine functional tabs (Home,

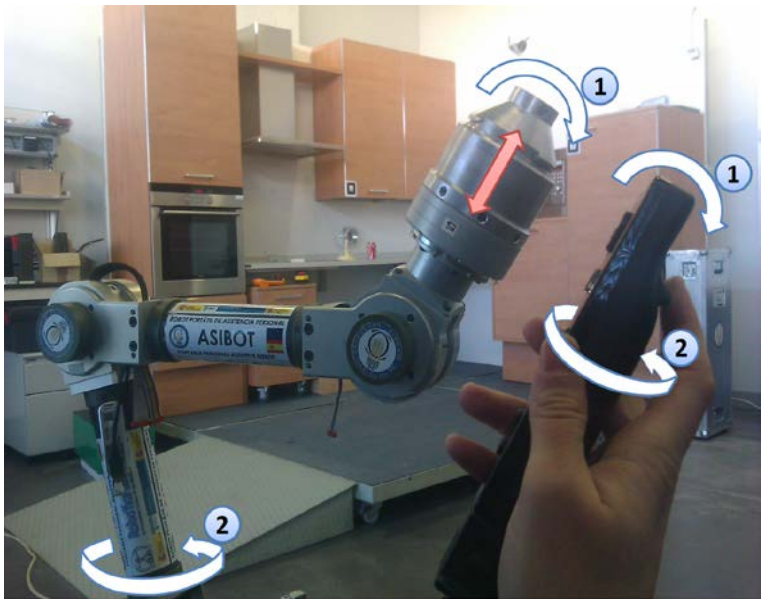
(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

Joint, Cartesian, Program, Speech, Assigner, Launcher, Video, and Docking). A persistent Connection Manager for establishing and terminating communications with the real robot and with the simulator is set to be rendered at the bottom left corner of the browser window. The client side scripts of the pages served have been optimized to minimize the amount of client-server interactions that take place.

## Wii Remote controller integration

A Wii Remote Plus controller interface module has been developed as part of the robot's open architecture components for multi-modal interaction. The controller's A and B buttons control forward and backward movement functionalities respectively, while maintaining both buttons pressed allows plain reorientation. The robot tip aligns with the Wii-Remote controller pitch, and the robot's base roll is controlled with the controller roll (Fig. 2).

*Figure 2. The Wii Remote orientation is tracked with a fixed linear velocity*

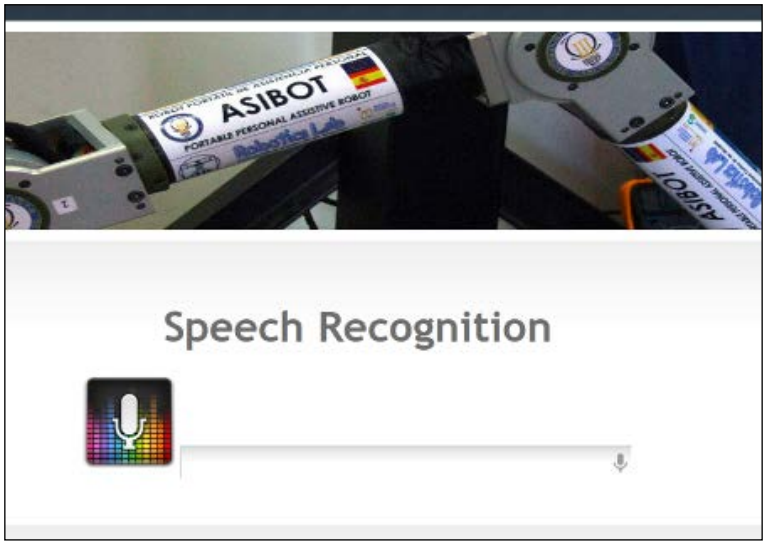


(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

## Automatic Speech Recognition

The robot's automatic speech recognition has been integrated into the Web Browsable interface as a selectable tab. The page served (Fig. 3) contains a speech recognition input field for recording and saving commands which can later be assigned to different tasks.

*Figure 3. The robot modules provide speech recognition for HRI*




The input field makes use of the x-webkit-speech attribute, linking the field to the Google Inc. implementation of the HTML5 Speech Input API (currently a W3C Editor's Draft (Sampath & Bringert, 2010)) by default. The Google Inc. implementation of the x-webkit-speech attribute uses Google's service cloud to perform the actual speech recognition, which returns a plain text string that the robot stores in its User Program Repository.

If the final system is not going to have access to a constant Internet connection, a local, but more limited, speech recognition mechanism may be used. This solution is based on PocketSphinx, part of the CMU Sphinx - Speech Recognition Toolkit (Huggins-Daines et al., 2006). This software is more accurate when a reduced dictionary (or 'corpus') of words is used. The corpus used is formed by common relevant words, including: color names, daily life activities (give, bring, wash, etc.), pronouns, and domestic objects (can, water, fridge, door, etc.). Once the corpus language model has been

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

created, the use is the same as if it were using the Google service cloud alternative. This local method is very robust to pronunciation and external noise, but lacks flexibility because its results are limited to the words contained within the developed corpus.

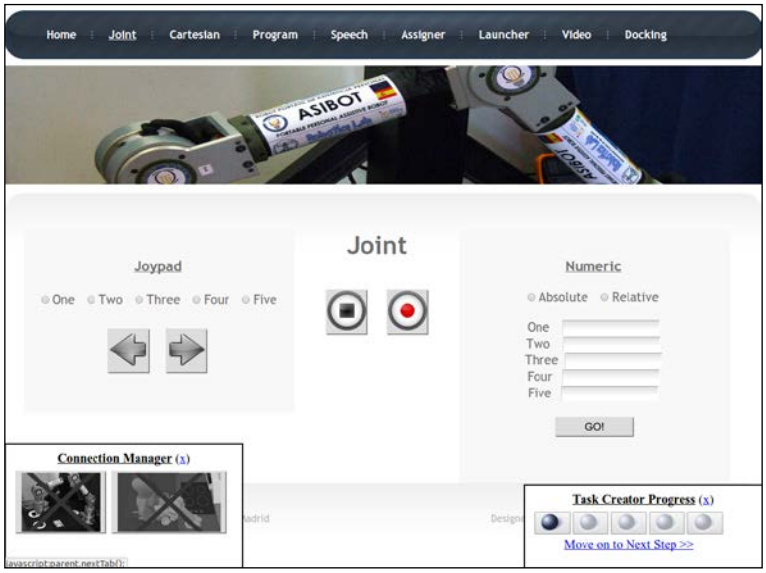
## Task Creator Wizard

The robot Web Browsable Task Creator Wizard has been developed to guide the user through the task creation process from within the robot Web Browsable interface. A robot task is composed by one or several custom or predefined programs that the user may invoke through the use of one or more of the open architecture's multi-modal interfaces. The Task Creator Wizard is initialized from within the Web Browsable interface homepage. It is set to display useful user guide information in the form of prompts and alerts. The use of the Wizard is, however, not mandatory. The user may instead choose to browse through the tabs manually to develop robot tasks.

First, once activated, the Wizard automatically redirects the user to the Joint space movement tab (Fig. 4). The tab is invoked so that a progress bar is displayed on the bottom right corner of the page. It indicates how advanced the user is in the task creation process, and allows the user to jump to each next step throughout the entire creative process.

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

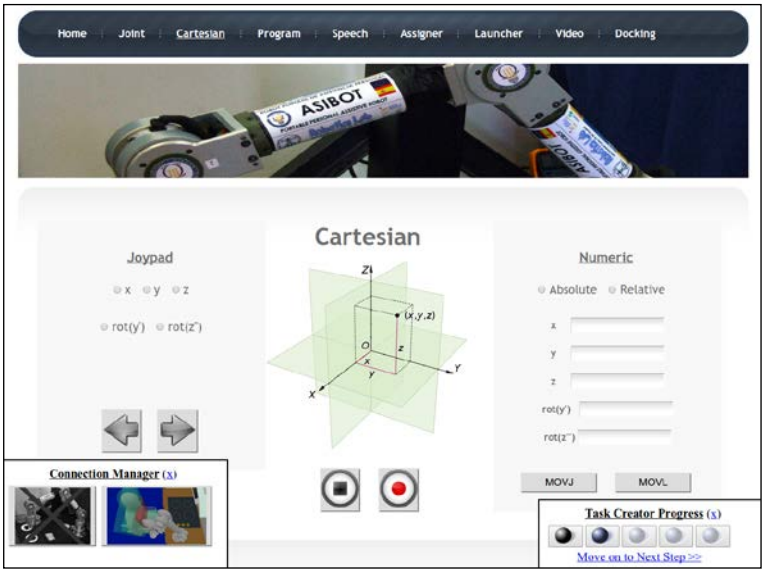
*Figure 4. A progress bar guides the user throughout the whole creative process*




As previously mentioned, the user can establish connections with the real and the simulated robot using the persistent Connection Manager situated at the bottom left of the interface. Once the connections are established, the user can move the selected robots in the Joint space using the correspondent tab buttons.

Additionally, the user can press the capture button (the round red Record icon situated at the center-right of the same Fig. 4) to open a prompt for saving the robot's tip point with a custom name. The robot's tip point position and orientation information that is stored is computed when the user clicks on the capture button, which may occur even if the robot is in movement. This behavior has also been implemented in the Cartesian space movement tab (Fig. 5), which is the next step the user is guided through.

*Figure 5. Points may be captured even when the robot is in movement*




The capture button of either of these two tabs, namely the Joint space movement tab and the Cartesian Space movement tab, may additionally be used to capture points when the robot is moved by using the Wii Remote Plus controller interface. On the completion of this point capturing phase of the Task Creation process, the user is guided by the Wizard to the Program tab (see Fig. 6). Here, the user can create, edit, save and delete robot programs directly from within the Web Browsable interface.

The robot Web Browsable interface Program tab plays the role of an Integrated Development Environment (IDE) for developing robot user Python programs. The left side panel allows the user to create, explore and delete robot user Python programs. When the user decides to create a new program, the IDE returns a new file with a snippet of default source code. This source code is extracted from a template file which is set to load the basic resources for programming the robot (libraries, initialization routine calls). Additionally, some hint lines of code are added for connecting to a remote instance of the robot or simulation control module, performing a robot homing movement, waiting, and closing the module cleanly.

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

*Figure 6. The robot Web Browsable interface Program Tab*


The Program tab additionally provides a set of buttons with the captured point names, situated on the top part of its right side panel. Clicking on this type of button inserts two lines of code into the central program text area:

- A Point definition. The definition of the point that was captured and given the name that the button indicates.
- A Cartesian space movement command. The robot is commanded from its current position to the point indicated, following a straight line trajectory (a MOVL command).

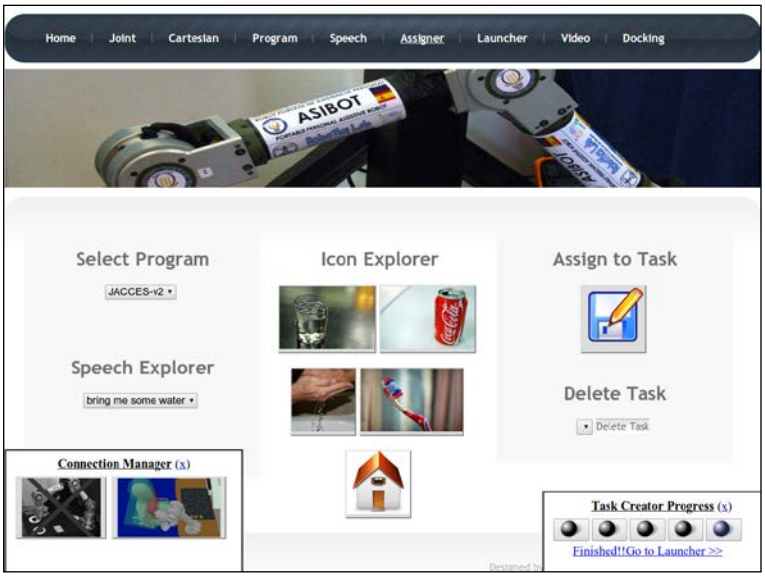
A robot Python point is defined as a native Python list of doubles that indicate the position and orientation of the robot tip in absolute base coordinates. The MOVL member function calls may also be modified and transformed into MOVJ function calls. Movements due to MOVJ function calls are, generally speaking, faster but less precise (trajectory-wise) than those issued by MOVL commands. This is because MOVL commands involve the computation of a straight linear trajectory, whereas MOVJ commands involve trajectory interpolation at single joint level. This nomenclature is commonly found in the context of industrial robots, and the authors have particularly been inspired by the RAPID, an ABB proprietary programming language (ABB, 2005).

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

Once the user has finished programming, she or he will be prompted to save the program with a custom name by pressing the save button. The Wizard then guides the user to the Speech tab. In the Speech tab, the user records and saves words that will be assigned to programs in the Task Creator final step, the Assigner tab (seen in Fig. 7). The Assigner tab is composed by program, recorded word, and icon selectors to generate robot task files, which are minimalistic scripts that link these three elements.

The Task Creator Wizard leads the user to the robot Web Browsable interface Launcher tab once the assignment has been performed. The Launcher parses the task files and presents the selected icons zoomed as touch buttons on screen, waiting for user tactile interaction or voice commands to execute the tasks that the user has developed through the use of the multi-modal interfaces, with or without the use of the interface's Task Creator Wizard.

*Figure 7. The robot Web Browsable interface Assigner Tab*




## Results

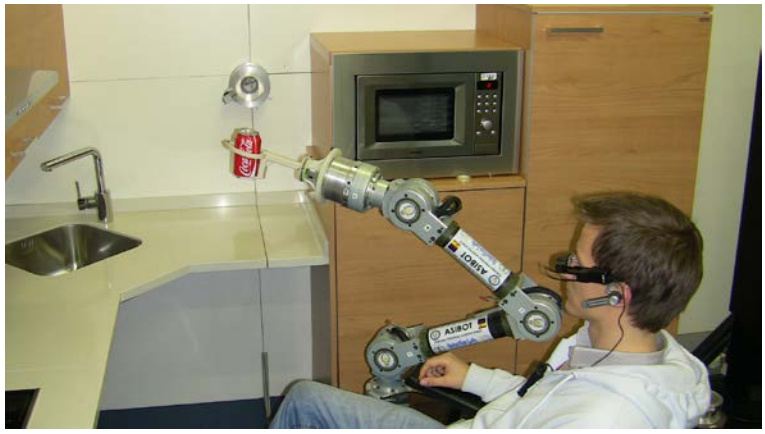
In order to perform a complete system assessment, we conducted two different tests: one with people that were previously inexperienced with robotics, and one with robotics-related people who were familiar with the

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

robot. The reason for this double-test was to include the not-so-common opinion of developers or technology-skilled users to the common analysis of inexperienced people. The comments and suggestions of technology-skilled users can be useful to assure an easy teaching process, as at a certain point they could actually become the people in charge of training disabled people in handling high-tech adapted devices.

In the first test, the ten healthy inexperienced users in robotics were invited to the robot assistive living kitchen environment and attended a five-day course for two hour sessions each day. At the end of the entire course, where they were explained how the system and the robot work, they were asked to use the developed interface presented for the creation of a common domestic task: grabbing a red can from a table. This is a task which we already knew that the robot was capable of performing (Fig. 8).

*Figure 8. The robot "Grab a red can from a table" task achievement*



To evaluate their experiences, we performed spoken interviews at the end of the five-day course. This type of feedback (instead of regulated tables or forms) was chosen because these non-experienced people found it easier to express their sensations by speaking naturally. They were all asked the same set of questions, about pros and cons, comfortability, complexity, and main drawbacks. The following is a summary of the answer received:

- From a domestic point of view, all of the users found the use of the proposed multi-modal interfaces (touch buttons, voice commands, and automatic speech recognition) very interesting and useful.
- Each of the users found a device that best fit their necessities to use the Web Browsable interface for comfortably interacting with the assistive robotic arm.
- All of the users were capable of generating several voice patterns that could be recognized by the Internet version of the automatic speech recognition system.
- Every user was able to make the robot grab the red can from the table successfully. From our professional experience, similar courses with industrial robots and controllers indicate a time closer to two weeks for performing a similar task.
- Two users evaluated the programming performed with the Wii Remote controller negatively. The reason was having to sustain the implemented "dead man"' buttons while moving the controller. They found it uncomfortable and counterintuitive.

In the second test, we asked ten robotics researchers to perform the same task, without the five-day course, and only a brief introduction to the system. To measure their satisfaction with the system, we provided them with standard SUS tests (System Usability Scale). The reason behind this difference in tests between robotics and non-robotics people is because, with the technological people, we were aiming toward system improvements beyond those that can be pointed out by inexperienced users. The following are the SUS results:

- The total average punctuation of the system was 70.5 ± 9.5 over 100 (where 100 is the best score).
- The best results were obtained in the statement: "I think that I would like to use this system frequently". Its numerical result was 4 ± 0.6 over 5 (where 5 is the best score).

(CC) JACCES, 2014 - 4(3): 161-176. ISSN: 2013-7087

- The worst results were obtained in the statement: “I think that I would need the support of a technical person to be able to use this system". Its numerical result was 2.6 ± 0.8 over 5 (where 5 is the worst score).

As a general overview, experiences were positive and the feedback received from robotics people indicates us where to improve the programming for a lighter and faster browsing interaction.

## Conclusions

The presented interface has proved to be a feasible and useful way to program an assistive robot for common activities in a domestic environment. Its multi-modal tools and its commercial off-the-shelf device integration (e.g. Wii Remote controller) enable the possibility of interacting with robots in a different way, which may result more accessible for certain people with special necessities. We ensure accessibility for all kinds of people, taking into account the fact that industrial robots are known for their use of their own, expensive and complex, programming elements. The presented system may be accessed from any kind of device with Internet capabilities.

The Task Creator Wizard ensures a complete process of programming a complex action, without having any knowledge of programming. This is a major advance towards the domestic introduction of advanced robotics. Additionally, the use of visually relevant icons helps to easily recognize pre-recorded tasks. The feedback received from the users has helped us understand where to focus future research efforts. The presented system gets closer to the original aim of the assistive robotic arm, which is to aid disabled and elderly people.

## Acknowledgments

The research leading to these results has received funding from the ARCADIA project DPI2010-21047-C02-01 granted by CICYT.

## References

[1] ABB. (2005). Rapid reference manual system data types and routines on-line. For BaseWare OS 3.1

[2] Burnett, Margaret M. & Scaffidi, C. (2011). End-user development. in Encyclopedia of Human-Computer Interaction.

[3] Diankov, R. (2010). Automated construction of robotic manipulation programs. Carnegie Mellon University.

[4] Fitzpatrick, P., Metta, G., & Natale, L. (2008). Towards long-lived robot genes. Robotics and Autonomous systems, 56(1), 29-45.

[5] Guizzo, E., & Ackerman, E. (2012). The Rise of the ROBOT WORKER. Spectrum, IEEE, 49(10), 34-41.

[6] Guo, C., & Sharlin, E. (2008, April). Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 121-130). ACM.

[7] Holzinger, A. (2003). Finger instead of mouse: touch screens as a means of enhancing universal access. In Universal Access Theoretical Perspectives, Practice, and Experience (pp. 387-397). Springer Berlin Heidelberg.

[8] Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnicky, A. I. (2006). Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006

Proceedings. 2006 IEEE International Conference on (Vol. 1, pp. I-I). IEEE.

[9] Iocchi, L., Ruiz-del-Solar, J., & Van der Zant, T. (2012). Domestic Service Robots in the Real World. Journal of Intelligent & Robotic Systems, 66(1), 183-186.

[10] Kim, M., Oh, K., Choi, J., Jung, J., & Kim, Y. (2011). User-Centered HRI: HRI Research Methodology for Designers. In Mixed Reality and Human-Robot Interaction (pp. 13-33). Springer Netherlands.

[11] Rode, J., Rosson, M. B., & Qui, M. A. P. (2006). End user development of web applications. In End User Development (pp. 161-182). Springer Netherlands.

[12] Sampath, S. & Bringert, B. (2010). Speech input api specification. W3C Editor's Draft 18.

[13] Victores, Juan G. (2010). Software engineering techniques applied to assistive robotics: Guidelines & tools. Master thesis. Dept. Syst. Eng. Autom. Univ. Carlos II of Madrid, Spain.

Fundación ONCE

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Accessibility Chair

# JACCES

ISSN: 2013-7087

Twitter: @Journal_JACCES LinkedIn: JACCES

www.jacces.org